Mirosław ŁAWRYNOWICZ, Jerzy JÓZEFCZYK
Politechnika Wrocławska

ANALIZA WARUNKÓW ROZWIĄZYWALNOŚCI PROBLEMÓW SZEREGOWANIA ZADAŃ Z NIEUSTALONYMI TERMINAMI GOTOWOŚCI

Streszczenie. Przedmiotem pracy jest zdefiniowanie oraz szczegółowa analiza warunków, które umożliwiają rozwiązanie problemów szeregowania zadań z nieustalonymi terminami gotowości na maszynach dowolnych w czasie wielomianowym. Zaproponowane modele bazują na kryterium długości uszeregowania i kryterium żalu dla długości uszeregowania (optymalizacja odporna). Zbadano jak nieustalone terminy gotowości zadań mogą wpływać na złożoność obliczeniową wybranych problemów.

ON THE SOLVABILITY OF JOB SCHEDULING PROBLEMS WITH NON-FIXED RELEASE DATES

Summary. The subject of this paper is a detailed analysis of the polynomial-time solvability of job scheduling problems with non-fixed release dates. Proposed models involve the makespan criterion and the minmax regret criterion for the makespan (robust optimization). We investigated how non-fixed release dates can affect the complexity of chosen problems.

1. Introduction

Analysis of an instance structure may allow to optimally solve an optimization or decision problem. Narrowing down decision space depends on the values of constituent variables. This type of research has a major impact on developing efficient exact algorithms when conditions for polynomial solvability are satisfied. Identifying these conditions for job scheduling with non-fixed release dates on unrelated machines is the main subject of this paper. By term “non-fixed” we mean the machine-dependent release dates or robust case in which each release date belongs to a well-defined interval. Namely, our aim is to investigate how the input parameters affect the hardness of underlying scheduling problems with the makespan criterion and its non-deterministic counterpart with the minimax regret criterion. Identification of polynomially solvable cases has been conducted for well-known optimization problems. In [8], the authors consider the single machine scheduling problem to minimize maximum weighted absolute deviations of job completion times from a common due-date. It is proven that the case of unit processing time jobs and the case of due-date assignment for a given job sequence ensures polynomial solvability. In [1], the authors show that the pooling

problem with one pool and a bounded number of inputs can be solved in polynomial time. Similar scientific papers in this field papers are [2], [11], [12].

Machine-dependent release dates result from the assumptions introduced in the Scheduling-Location (ScheLoc) problem. In ScheLoc problems the objective is to find locations for the machines and a schedule for each machine subject to some production and location constraints such that some scheduling objective is minimized or maximized [4], [10]. Then, each release date depends on the distance between a job's location and machine's location. Consequently, a vector of unequal release dates is defined for each job. For the makespan criterion and arbitrarily selected locations for machines, the ScheLoc problem comes down to $R|r_{i,j}|C_{\max}$. Only [7] deals with the machine-dependent release dates. The two constructive algorithms are developed for $R|r_{i,j}|C_{\max}$. It is proven that a polynomial-time greedy algorithm with a long-term planning guarantees an approximation ratio depending on the unrelated machines' performance. A second algorithm uses a simple decomposition strategy to solve optimally and sequentially subproblems for a predefined number of jobs. This approach is based on a brute-force method. Imprecise coordinates of locations for machines require a robust decision-making. This case forces imprecise release dates as shown in [6]. To handle this case the minimax regret version of $R|r_{i,j}|C_{\max}$ with interval release dates, denoted by $R|r_{i.j}^{-} \le r_{i,j} \le r_{i.j}^{+}|\mathrm{Reg}(C_{\max})$, has been investigated in [5] where the comparison between a greedy algorithm and Tabu Search metaheuristic has been conducted.

The paper starts with the formulations of $R|r_{i,j}|C_{\max}$ and $R|r_{i.j}^{-} \le r_{i,j} \le r_{i.j}^{+}|\mathrm{Reg}(C_{\max})$ in Section 2. The central part of the considerations given in Section 3 refers to the analysis of polynomial solvability of the scheduling problems. Finally, the concluding remarks are given in Section 4.

2. Scheduling problems

Denote by $M=\{1,2,...,j,...,n\}$ the set of *n* independent, non-preemptive jobs to be processed on the set $M=\{1,2,...,i,...,m\}$ of *m* unrelated machines. A job release date $r_{i,j} \ge 0$ and job processing time $p_{i,j} > 0$ depend on machine i. All the parameters are stored in the two-dimensional matrices $p=\left[p_{i,j}\right]_{\substack{i=1,2,...,m\\ j=1,2,...,n}}$, $r=\left[r_{i,j}\right]_{\substack{i=1,2,...,m\\ j=1,2,...,n}}$, respectively. The complete schedule is represented by $x=\left[x_{i,k,j}\right]_{\substack{i=1,2,...,m\\ k,j=1,2,...,n}}$ where $x_{i,k,j} \in \{0,1\}$ that indicates if job j is assigned to the *k*th, $k=1,2,...,n$, position in a sequence deployed on i. The *k*th job completion time on i can be calculated by the equation

$$C_{i,k}(x;S)=\sum_{j=1}^{n} x_{i,k,j}\left(p_{i,j}+\max\left\{C_{i,k-1}(x;S),r_{i,j}\right\}\right),\ i=1,2,...,m,\ k=1,2,...,n,\quad C_{i,0}(x;S)=0, \tag{1}$$

and $\underset{\substack{i=1,2,...,m\\ k=1,2,...,n}}{\forall} C_{i,k}(x,r)=0$ if $x_{i,k,j}=0$, $S=\{p,r\}$, and the makespan is determined by

$$C_{\max}(x;S)=\max_{i=1,2,...,m} C_{i,n_i}(x;S),\ n_i=\sum_{k=1}^{n}\sum_{j=1}^{n} x_{i,k,j}.$$

To sum up: for given J, M, R, p, the deterministic scheduling problem $R \mid r_{i,j} \mid C_{\max}$ consists in the determination of the optimal matrix x^* to minimize

$$C_{\max}\left(x^*;S\right)=\min_{x} C_{\max}\left(x;S\right) \tag{2}$$

subject to

$$\sum_{i=1}^{m}\sum_{k=1}^{n} x_{i,k,j}=1, \qquad j=1,2,...,n, \tag{3}$$

$$\sum_{j=1}^{n} x_{i,k,j}\le 1, \qquad i=1,2,...,m, \;\; k=1,2,...,n, \tag{4}$$

$$\sum_{j=1}^{n} x_{i,1,j}\le m, \qquad i=1,2,...,m, \;\; k=1,2,...,n, \tag{5}$$

$$x_{i,k+1,j}-x_{i,k,j'}\le 0, \qquad i=1,2,...,m, \;\; k=1,2,...,n-1, \;\; j\ne j', \tag{6}$$

$$x_{i,k,j}\in\{0,1\}. \tag{7}$$

Each job has to be assigned only once according to (3). At most one job can be handled in a single position as specified in (4). Not every machine has to be used that is allowed by (5). A sequence of jobs deployed on a single machine has to be represented by the consecutive non-zero entries which is stated in (6). Such a technical constraint (6) guarantees the correctness of (1) and improves the readability of the schedule. Finally, the binary decision variable domain is given in (7).

The non-deterministic version of (2) involves imprecise release dates. It is assumed that release date of job j on machine i belongs to the well-defined interval $u_{i,j}=\left[r_{i,j}^{-},r_{i,j}^{+}\right]_{\substack{i=1,2,...,m\\ j=1,2,...,n}}$, $0\le r_{i,j}^{-}\le r_{i,j}^{+}$. Then, the set of feasible release dates of job j is determined by the Cartesian product of intervals $U_j=u_{1,j}\times u_{2,j}\times...\times u_{m,j}$. All the feasible scenarios are stored in $U=U_1\times U_2\times...\times U_n$. The decision variable has the same form as (7) and is denoted by $\tilde{x}$. Then, a value of the maximal regret is equal to $\max_{u\in U}\left\{C_{\max}\left(\tilde{x}\;u\;S\right)-C_{\max}\left(\tilde{x}\;u\;S\right)\right\}$.

To sum up: for given J, M, U, p, the non-deterministic scheduling problem $R \mid r_{i.j}^{-}\le r_{i,j}\le r_{i.j}^{+} \mid \mathrm{Reg}\left(C_{\max}\right)$ deals with the determination of the optimal matrix $\tilde{x}$ to minimize

$$Q\left(\tilde{x}\right)=\max_{u\in U}\left\{C_{\max}\left(\tilde{x}\;u\;S\right)-C_{\max}\left(\tilde{x}\;u\;S\right)\right\}. \tag{8}$$

subject to (3)-(7).

3. On the solvability of $R|r_{i,j}|C_{\max}$ and $R|r_{i.j}^{-} \le r_{i,j} \le r_{i.j}^{+}|\mathrm{Reg}(C_{\max})$

The complexity of $R|r_{i,j}|C_{\max}$ results from the complexity of its simpler version $P|C_{\max}$. For $\underset{i\neq s, j\neq t}{\forall} r_{i,j} = r_{s,t}$ and $\underset{j\neq t}{\forall} p_{i,j} = p_{i,t}$, the problem (3) comes down to $P|C_{\max}$ even if $r_{i,j} \neq 0$, which is strongly NP-hard for at least three machines and at least NP-hard for exactly two machines [3]. Next we will discuss the conditions of polynomial solvability of $R|r_{i,j}|C_{\max}$ and $R|r_{i.j}^{-} \le r_{i,j} \le r_{i.j}^{+}|\mathrm{Reg}(C_{\max})$. The chosen cases are listed below.

1) *Impact of a single release date on the solvability of* $R|r_{i,j}|C_{\max}$.

Although the simplicity of the proof of NP-hardness of $R|r_{i,j}|C_{\max}$, establishing connections between the processing times and release dates sheds more light on the problem's complexity. Since the matrices p and r are independent, a seemingly insignificant change of some values may have an impact on the decision space. A considerable improvement in a single machine performance may facilitate the problem. For instance, if

$$\underset{j=1,2,\ldots,n}{\forall}\left(i = \arg\min_{i'=1,2,\ldots,m} p_{i,j} \ \wedge\ i = \arg\min_{i'=1,2,\ldots,m} r_{i,j}\right) \ \wedge\ \underset{j\neq j'}{\exists} r_{i,j} + p_{i,j} \le r_{i,j'}\ , \tag{9}$$

the optimal sequence is deployed on i irrespective of remaining machines' $(M \setminus i)$ parameters when for job j exists unique j' such that $\underset{j\neq j'}{\exists} r_{i,j} + p_{i,j} \le r_{i,j'}$ excluding the latest available job. This solution can be achieved using the Earliest Release Date (ERD) rule [9] that assigns the jobs as soon as there is machine availability to process them.

2) *Impact of a single machine's parameters on the solvability of* $R|r_{i,j}|C_{\max}$.

Let us consider another case in which the release dates of job *j* are delayed in such a way that

$$\max_{\substack{i'\in M\\ j'\in J\setminus j}} r_{i',j'} + \sum_{j'\in J\setminus j} \max_{i'\in M} r_{i',j'} \le r_{i,j}, \qquad i = 1,2,\ldots,m. \tag{10}$$

Then a value of (2) depends only on the parameters of job *j* if the remaining jobs are scheduled before *j*. In fact, each instance could be easier to solve if release date of a single job can be modified. Notice that an order of $J \setminus j$ does not affect the makespan irrespective of a schedule constructed before $r_{i,j}$. As a result, the optimal value of (2) is achieved if *j* is assigned to the machine

$$i' = \arg\min_{i'=1,2,\ldots,n}\left(r_{i',j} + p_{i',j}\right) \tag{11}$$

in the last position of a sequence.

3) *Impact of a single machine's parameters on the solvability of* $R|r_{i,j}|C_{\max}$.

Let us introduce a more sophisticated instance. We assume the jobs order of availability is the same on each machine but only a single machine $i \in M = \{1,2\}$ ensures that $r_{i,j} + p_{i,j}$, for unique job j, is lower than the subsequent jobs release dates as shown in Figure 1.

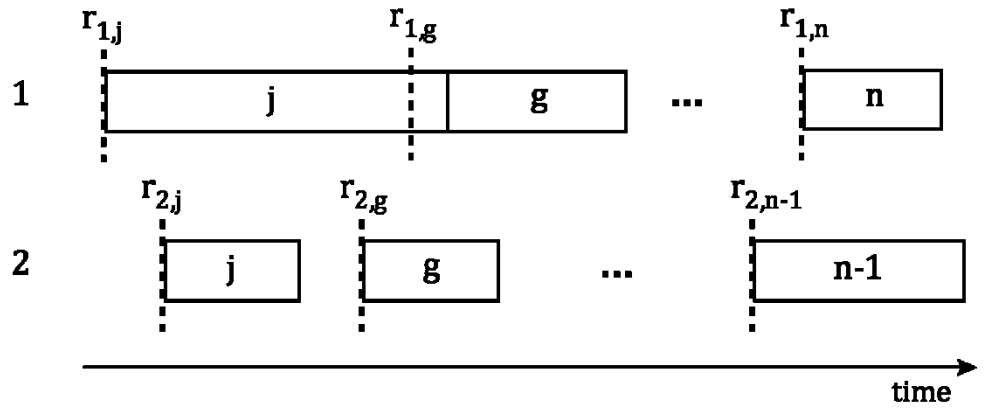


Fig. 1. The instance with two machines $M = \{1,2\}$ and n jobs where an optimal schedule determined for $J \setminus \{n, n+1\}$ does not affect the makespan, and $r_{i,n} = r_{i,n-1}$, $i \in M$

It is easy to see that the greedy algorithm solves the underlying instance optimally by sequential determination of $\arg \min_{i=1,2,...,m} \{r_{i,j} + p_{i,j}\}$ for each non-assigned job. Looking at Figure 1, remark that the makespan can be expressed using only the parameters of a single job if a schedule is optimal.

4) *Polynomial-time solvability of* $R \mid r_{i.j}^{-} \le r_{i,j} \le r_{i.j}^{+} \mid \mathrm{Reg}(C_{\max})$.

It turns out that conditions (9) and (10) defined for $R \mid r_{i,j} \mid C_{\max}$ ensure polynomial solvability of its non-deterministic counterpart $R \mid r_{i.j}^{-} \le r_{i,j} \le r_{i.j}^{+} \mid \mathrm{Reg}(C_{\max})$ although the evaluation of $C_{\max}(\tilde{x} \; u \; S')$ in (8) requires solving $R \mid r_{i,j} \mid C_{\max}$ optimally, which is at least NP-hard. Remark that a value of (8) is greater than zero only if some subset of jobs scheduled according to $\tilde{x}$ can be handled earlier in $\tilde{x}$. Hence, if condition (9) is satisfied then any job cannot be rescheduled in $\tilde{x}$ because the completion time of each job is the earliest on the same machine and $Q(\tilde{x}'$. Let us modify (10) as follows:

$$\max_{\substack{i' \in M \\ j' \in J \setminus j}} r_{i',j'} + \sum_{j' \in J \setminus j} \max_{i' \in M} r_{i',j'} \le r_{i,j}^{-}, \qquad i = 1,2,...,m. \tag{12}$$

If condition (10) is satisfied for j that is scheduled on (11) then the makespans $C_{\max}(\tilde{x} \; u \; S'$ and $C_{\max}(\tilde{x} \; u \; S'$ are equal because only $j' \in J \setminus j$ can be rescheduled.

We will show that the generalization of the case described in 3) for *m* machines determines such a new scheduling problem that release dates and deadlines are machine-dependent. In general, if there exist m jobs in

$$L(r,J)=\left\{ j\in J \mid j=\arg\max_{l=1,2,...,n} r_{i,l} \text{ for } i=1,2,...,m \ \wedge\forall_{j\neq j'} r_{j,l}\neq r_{j',k}\wedge\forall_{k\neq l} r_{j,l}=r_{j,k}\right\} \quad (13)$$

and we are able to create a schedule of the jobs in $J\setminus L(r,J)$ such that

$$C_{i,n_i}(x;S)<r_j,\ j\in L(r,J),\ \left|L(r,J)\right|=m,\ i=1,2,...,m, \quad (14)$$

then the makespan depends only on the jobs in (13). r_j refers to machine-independent release date. Thus traceability results from scheduling *m* jobs on *m* machines. For this reason, condition (14) narrows down a subset of polynomial time solvable cases. Clearly, the parameters in S can be adjusted for any matrix *x* in polynomial time to satisfy (14). On the other hand, we will prove that the complexity of verifying condition (14) for an undefined *x* is computationally hard. Solving this problem would enable us to identify a traceable instance. First, let us impose the following constraints (see Figure 2) on the instance of (2):

$$\left|L(r,J)\right|=2, \quad (15)$$

$$\forall_{e\in J\setminus\{J_p\cup J_a\}} p_{1,e}=p_{2,e},\ J_p=\{j,l\},\ J_a=\{g,h\}, \quad (16)$$

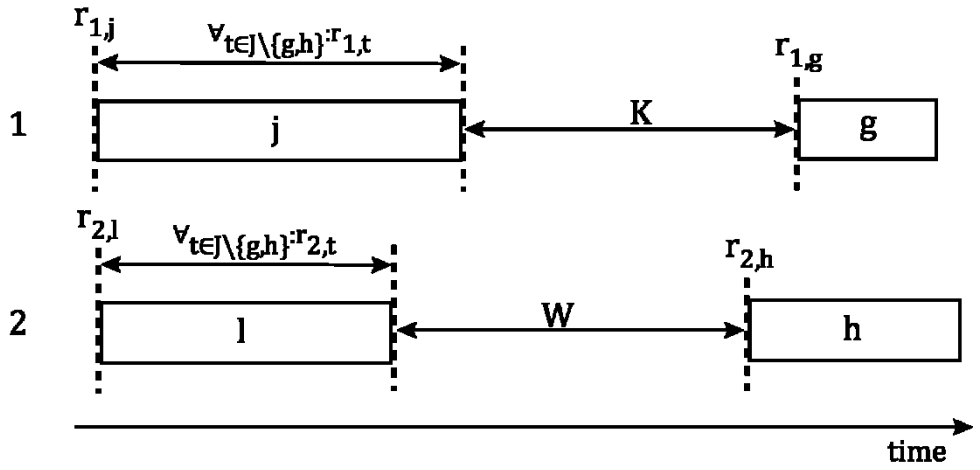
$$r_{w,t}=\min_{j=1,2,...,n} r_{w,j},\ p_{w,j}=p_{w,l},\ t\in J_p,\ w\in\{1,2\}, \quad (17)$$

$$p_{w,t}>\sum_{e\in J\setminus\{J_p\cup J_a\}} p_e \text{ and } r_{w,t}+p_{w,t}>r_{w,b},\ t\in J_p,\ b\in J_a,\ w\in\{1,2\}, \quad (18)$$

$$\text{K+W}=\sum_{J\setminus\{J_p\cup J_a\}} p_e \text{ and K>W}. \quad (19)$$

The latest release date on each machine refers to a unique job (15). The processing times of the jobs in $J\setminus\{j,l,g,h\}$ are not machine-dependent (16). Both j and l has the identical parameters on each machine (17). Constraint (18) stipulates that the release date of each job in $J\setminus\{j,l,g,h\}$ is covered by the processing times of j or l. Term (19) describes the instance-specific conditions. Clearly, this instance can be constructed in polynomial time.

The decision problem referred to as **Condition** is as follows: *Does exist a schedule such that* (14) *is satisfied ?*
We will prove the NP-completeness of this decision problem via the transformation from **Subset Sum** problem [3]. Let us begin by presenting its formal description.
Subset Sum
Instance: Finite set B and given size $s(b)>0$ for each $b\in B$, positive integer K.
Question: Is there a subset $B'\subseteq B$ such that the sum of the sizes of the elements in B' is exactly K ?



The polynomial transformation of **Subset Sum** problem to the instance of **Condition** is based on (15)-(19), $\tilde{J}$ J $\{j,l,g,h\}$, $M=\{1,2\}$, the release dates $r_{i,t}\geq 0$, and the processing times $p_t>0$, $t\in\tilde{J}$.

Theorem 1. **Condition** *is NP-complete.*

Proof. It is evident that the condition (13) can be verified in polynomial time for the given x. Therefore, the decision problem is in NP because there exists a poly-time certifier for (13). The local replacement assumes $t=b$, $p_t=s(b)$, $\tilde{J}$ B.

To ensure (13), the jobs must be scheduled within the unused time slots K and W, as illustrated in Figure 2. This can be done if and only if there exists a subset $\tilde{J}$ $\tilde{J}$ such that $\mathrm{K}=\sum_{e\in\tilde{J}} p_e$. In effect, machine 2 must handle $\sum_{e'\in\tilde{J}\ \tilde{J}} p_{e'}=\mathrm{W}$ time units of processing within the available time slot due to (19). Hence the required subset $\tilde{J}$ exists for the instance of **Subset Sum** if and only if a feasible schedule exists for the corresponding instance of **Condition**. Q.E.D

3. Final remarks

Our research raises the subject of an impact of the values of input parameters on the complexity of $R\,|\,r_{i,j}\,|\,C_{\max}$ and $R\,|\,r_{i.j}^{-}\leq r_{i,j}\leq r_{i.j}^{+}\,|\,\mathrm{Reg}(C_{\max})$. The proposed conditions demonstrate rather strikingly that the long time gaps between the release dates on each machine simplifies the problem. The essential point to note here is that balancing the load over the various machines, which is an important goal in practice, may not occur in some instances. Moreover, we indicated that the conditions of polynomial solvability of $R\,|\,r_{i,j}\,|\,C_{\max}$ and $R\,|\,r_{i.j}^{-}\leq r_{i,j}\leq r_{i.j}^{+}\,|\,\mathrm{Reg}(C_{\max})$ can be equivalent. We proved that a simple condition can determine the new scheduling problem that has not been yet studied in the literature. These considerations give some insights into the structure of chosen scheduling problems and demonstrate the difficulty in generating relevant datasets, which is a crucial topic in numerical experiments.

REFERENCES

1. Boland, N., Kalinowski, T., & Rigterink, F. (2017). A polynomially solvable case of the pooling problem. Journal of Global Optimization, Vol. 67 No. 3, p. 621-630.
2. García, A., &Tejel, J. (2017). Polynomially solvable cases of the bipartite traveling salesman problem. European Journal of Operational Research, Vol. 257 No. 2, p. 429-438
3. Garey, M. R., & Johnson, D. S. (1979). Computers and intractability (Vol. 174). San Francisco: freeman.
4. Heßler, C., & Deghdak, K. (2017). Discrete parallel machine makespan ScheLoc problem. Journal of Combinatorial Optimization, Vol. 34 No. 4, p. 1159-1186.
5. Józefczyk J., & Ławrynowicz M. (2021), On selected models and methods of robust decision-making and their applications, World Organisation of Systems and Cybernetics (WOSC Congress)
6. Józefczyk, J., Ławrynowicz, M., & Filcek, G. (2022) On problems and methods of coordinated scheduling and location. W: Uncertainty and Imprecision in Decision Making and Decision Support: New Advances, Challenges, and Perspectives : Selected papers from BOS-2020, held on December 14-15, 2020, and IWIFSGN-2020, held on December 10-11, 2020 in Warsaw, Poland / eds. Krassimir T. Atanassov [i in.]. Cham : Springer, cop. 2022. s. 145-163. (Lecture Notes in Networks and Systems, ISSN 2367-3370; Vol. 338)
7. Ławrynowicz, M., & Józefczyk, J. (2022, August; Accepted). Scheduling jobs with machine-dependent release dates on unrelated machines. In 2022 26th International Conference on Methods and Models in Automation and Robotics (MMAR) IEEE.
8. Mosheiov, G., & Sarig, A. (2010). Scheduling with a common due-window: polynomially solvable cases. Information Sciences, Vol. 180 No. 8, p. 1492-1505.
9. Pinedo, M. L. (2012). Scheduling (Vol. 29). New York: Springer.
10. Rajabzadeh, M., Ziaee, M., & Bozorgi-Amiri, A. (2016). Integrated approach in solving parallel machine scheduling and location (ScheLoc) problem. International Journal of Industrial Engineering Computations, Vol. 7 No. 4, p. 573-584
11. Smet, P., Brucker, P., De Causmaecker, P., & Berghe, G. V. (2016). Polynomially solvable personnel rostering problems. European Journal of Operational Research, Vol. 249 No. 1, p. 67-75
12. Zukerman, M., Jia, L., Neame, T., & Woeginger, G. J. (2001). A polynomially solvable special case of the unbounded knapsack problem. Operations Research Letters, Vol. 29 No. 1, p. 13-16